OIPE
JAN 2 1 2005
PATENT & TRADEMARK OFFICE

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:
Copeland et al.

Serial No. 09/740,460

Filed: December 18, 2000

For: JSP COMPOSITION IN A CACHE FOR WEB APPLICATIONS WITH DYNAMIC CONTENT

Group Art Unit: 2154
Examiner: Mohammad A. Siddiqi

Atty. Dkt. No. AUS9000460US1

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on the date indicated below:

January 18, 2005
Date

Kevin L. Daffer

APPEAL BRIEF

Box AF
Commissioner for Patents
Washington D.C. 20231

Sir/Madam:

Further to the Notice of Appeal faxed November 18, 2004 and received in the U.S. Patent and Trademark Office on the same day, Appellant presents this Appeal Brief. The Notice of Appeal was filed following mailing of a Final Office Action on September 2, 2004. Appellant hereby appeals to the Board of Patent Appeals and Interferences from a final rejection of claims 1-3, 5-15, 17 and 18 in the Final Office Action, and respectfully requests that this appeal be considered by the Board.

I. REAL PARTY IN INTEREST

The subject application is owned by International Business Machines Corporation, a corporation having its principal place of business at New Orchard Road, Armonk, New York, 10504, as evidenced by the assignment recorded at Reel 011391, Frame 0839.

II. RELATED APPEALS AND INTERFERENCES

No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-18 were originally filed in the present application. Claims 1, 8, 9, 16, 17 and 18 were amended in a response filed June 25, 2004 to an Office Action mailed March 25, 2004. Claims 14 and 16 were cancelled in the same response. Claims 1-3, 5-15, 17 and 18 stand finally rejected under 35 U.S.C. § 103, which are the subject of this appeal. A copy of claims 1-3, 5-15, 17 and 18, as on appeal, is included in the Appendix hereto.

IV. STATUS OF AMENDMENTS

Amendments to the claims have not been filed subsequent to their final rejection. The Appendix hereto therefore reflects the current state of the claims.

V. SUMMARY OF THE INVENTION

Appellant's claimed invention relates to a software system, computer program product, web server and method for caching a parent server page and a child server page on the web server side.

In some embodiments, the presently claimed software system may include a parent server page (e.g., JSP1 200, Fig. 5), containing a call to a child server page (JSP2), and a cache (154, Fig. 4) located within a web server (14, Fig. 1) and containing code for both the parent and child server pages. In preferred embodiments of the invention, however, the code for the parent server page does not contain all of the code for the child server page. Instead, the software system may include a link (*see,* e.g., the link within JSP1 206, Fig. 5), which is associated with the call to the child server page, and encapsulating information for locating and executing the code for the child server page. In some cases, the encapsulating information may include an instruction sequence that may be invoked to locate the child server page in the cache. However, if the child server page cannot be located within the cache, the child server page may be executed using the link, without having to re-execute the parent server page. In some cases, the link may include a web page address and a list of request attributes (Specification -- page 27, line 32 to page 29, line 5).

In some embodiments, the presently claimed method for caching a parent and a child server page may include: storing code for the parent server page in a cache located internal to a web server, such that the code for the parent server page does not contain all lines of code for the child server page; storing only one copy of the code for the child server page in the cache; creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page (Specification -- page 13, lines 4-18).

In some embodiments, the parent and child server pages may comprise Java Server Pages (JSP). In some cases, the presently claimed method may include invoking an instruction sequence to locate the code for the child JSP in the cache, in response to a request made by a web browser. If the child JSP cannot be located in the cache, the method may execute the code for the child JSP using the link, without executing all of the code for the parent JSP. In some cases, the child JSP may be executed in the web server in response to a request made by a client or another web server. By using the link, the parent and child JSPs may be cached separately, thus, allowing the cached child JSP to be updated without also updating the parent JSP (Specification -- page 29, lines 6-22).

VI. ISSUES

1. Whether claims 1-3, 5-15, 17 and 18 are unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,598,048 to Carneal et al. (hereinafter "Carneal") in view of U.S. Patent No. 6,643,652 to Helgeson et al. (hereinafter "Helgeson").

VII. GROUPING OF CLAIMS

Claims 1-3 and 5-8 (Group I) stand or fall together.
Claims 9-15, 17 and 18 (Group II) stand or fall together.

The reasons why the two groups of claims are believed to be separately patentable are explained below in the appropriate parts of the Argument.

VIII. ARGUMENT

Due to the tremendous growth in internet traffic, e-commerce websites are often hard pressed to deliver dynamic content in a prompt and efficient manner to the website user. One solution to the huge growth in internet traffic is to cache web content that is frequently requested, but infrequently modified, thereby avoiding the need to continually retrieve the content from the website database. In some cases, it may be preferred that the dynamic content be stored in an internal or external cache on the web server side (e.g., to enable statistics tracking for monitoring the number of users visiting a site). An "internal cache" may be described as located within the web server itself, while an "external cache" is one that is located anywhere between the web server and the internet boundary (i.e., on the web server side). Though caching of dynamic web content can improve the responsiveness of an e-commerce website without incurring the high cost of additional servers, caching performance typically depends on a number of factors including, but not limited to, cache capacity and efficiency. *See* Specification: page 1, line 22 to page 3, line 24; page 10, line 6 to page 11, line 15.

In some cases, Java Server Pages (JSP) and servlets may be used to generate the dynamic web content. JSPs and servlets can be nested – i.e., one JSP or servlet can call another. A JSP or servlet called by another JSP or servlet is sometimes referred to as "nested" or "embedded". In some cases, the main body of a (parent) JSP may contain one or more calls to other (child) JSPs. When a JSP containing calls to other JSPs is cached, a fully expanded version of each called JSP is usually embedded at the point in the main code from which it is called. This may result in the storage of multiple copies of the called (child) JSP, since it may be called from more than one place in the main (parent) JSP. In addition to reducing cache capacity, the above approach reduces cache efficiency when child JSPs are cached in fully expanded form within the parent JSP. If, for example, data or code within a child JSP changes in main memory, the cached version of the child JSP must be invalidated. However, since the child JSP is cached in expanded form within the parent JSP, the parent JSP must also be invalidated. Therefore, instead of merely updating the child JSP, the above approach reduces cache efficiency and diminishes system overhead by requiring that all cached content (including the parent JSP) be updated. *See* Specification: page 7, lines 3-18; page 29, lines 6-22.

Therefore, a need exists for a system and method that improves cache capacity by reducing the amount of cache space required to store "embedded" JSPs. Such a system and method would also improve cache efficiency by reducing the amount of cached content that has to be updated when a child JSP is invalidated. *See* Specification: page 28, lines 17-18 and page 29, lines 17-19.

The invention as recited in claims 1-3, 5-15, 17 and 18 addresses the above-described need for improved cache performance by providing a software system, computer program product, web server and method for caching parent and child server pages on the web server side. *See* Specification: page 13, lines 4-18; page 27, line 32 to page 29, line 22.

In some embodiments, the presently claimed software system may include a parent server page containing a call to a child server page, and a cache (located within a web server) containing code for both the parent server page and the child server page. To improve cache performance, the software system may further include a link, which is associated with the call to the child server page, and encapsulating information for locating and executing the code for the child server page. This allows the parent and child server pages to be cached separately, since the link may be used to locate the child server page in the cache, or to execute the child server page without having to re-execute the parent server page. In this manner, use of the presently claimed link reduces the number of cached lines (i.e., improves cache capacity) by <u>not</u> requiring that the parent server page contain all of the code for the child server page (as in the conventional manner, where the child server page is embedded in expanded form within the parent server page when cached). In addition, use of the presently claimed link improves cache efficiency by <u>not</u> requiring that the parent server page be updated when changes are made to the child server page, since the link preserves the connection to the new, updated version of the child.

In a more specific embodiment, the presently claimed computer program product, web server and method for caching a parent and a child server page may include: storing code for the parent server page in a cache located internal to a web server, such that the code for the parent server page does not contain all lines of code for the child server page. Unlike the conventional method, however, only one copy of the code for the child server page may be stored in the cache (instead of redundant, multiple copies of code for each call to the child server page). For example, a link to the singular copy of the child server page code may be cached along with the code for the parent server page; the link may later be used for locating and executing the code for the child server page. In some cases, the link may be associated with more than one call to the child server page. Even though the singular copy of the code for the child server page may be executed (from the cache) more than once, use of the presently claimed link improves cache performance by: (i) reducing the amount of cache space required to store "embedded" JSPs, and (ii) reducing the amount of cached content that has to be updated when a child JSP is invalidated.

As described in more detail below, none of the cited art, either separately or in combination, provides motivation, teaching or suggestion for the presently claimed system and method for caching parent and child server pages on the web server side. Therefore, the teachings of the cited art cannot be used to render the limitations of the presently claimed case unpatentable.

ISSUE 1 ARGUMENTS

Patentability of Group I Claims 1-3 and 5-8:

1. **Carneal fails to provide teaching or suggestion for a software system including a cache located within a web server and containing code for a parent server page and a child server page.**

Claim 1 recites, in part, "[a] software system supporting distributed web applications, comprising … a cache within a web server, containing code for the parent server page and child server page." In this manner, the Specification and present claims disclose a web content caching system where parent and child server pages are each cached on the web server side.

Carneal, on the other hand, fails to provide teaching or suggestion for storing parent and child server pages within a cache located within a web server. However, statements in the Final Office Action suggest that Carneal discloses "a cache within a proxy server (e.g., col. 7, lines 63-65), containing code for the parent server page (e.g., col. 8, lines 7-10) and child server page (e.g., col. 8, lines 7-10, an inline object is child page) …" (Final Office Action, page 3). The Examiner recognizes that Carneal discloses a cache located within a client-side proxy server, and admits that "Carneal fails to disclose a cache within a web server" (Final Office Action, page 3). The Appellants appreciate the Examiner's recognition of the lack of teaching within Carneal for storing parent and child server pages in a cache located within a web server. In addition, supporting evidence will be provided in more detail below (*See*, Argument 2) for why Carneal cannot be modified for caching parent and child server pages on the web server side.

Contrary to the above Office Action statements, Carneal fails to provide teaching or suggestion for a cache that contains code for both parent and child server pages, regardless of where the cache is located. In the passage cited by the Examiner (col. 8, lines 7-10), Carneal merely states that a parent file may contain references to an inline object (the alleged child server page). Though Carneal discloses that the inline objects may be cached (e.g., within object cache 71 of access point 70, Fig. 6), Carneal does not teach or suggest that the parent file may be cached along with the inline objects. Instead, Carneal specifically states that "after the [parent] file traverses the wireless link, the access point 70 forwards the parent file to the web browser 20" (*See*, Carneal, column 8, lines 22-28; Fig. 6). As such, Carneal only discloses that the inline objects may be cached (*See*, Carneal, column 8, lines 29-43) and provides no indication whatsoever that the parent file may also be cached.

2. The teachings of Carneal cannot be modified or combined with Helgeson to provide the claimed software system for caching parent and child server pages on the web server side.

As noted in the above Argument, the Examiner admits that "Carneal fails to disclose a cache within a web server" (Final Office Action, page 3). However, the Examiner suggests that the placement of the cache is a matter of design preference, and therefore, suggests that the teachings of Carneal could be modified to include a cache located in a web server (Final Office Action, pages 3-4). For example, the Examiner suggests that since "Helgeson ... disclose[s] a cache within a web server (col. 63, lines 65-67)," it would have been obvious to modify the teachings of Carneal with the web server side cache of Helgeson to provide control over caching in a web content server (Final Office Action, page 4). As set forth below, Appellants assert that there is absolutely no motivation within the teachings of Carneal that would enable one skilled in the art to make the proposed modification.

First of all, Carneal provides no teaching, suggestion or even desirability for caching both parent and child server pages in a cache located within the web server. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990); MPEP 2143.01. Appellants assert that the teachings of Carneal cannot be combined or modified with the web server side cache of Helgeson, since Carneal fails to even suggest a desirability for doing so. The mere mention of a cache within a web server (as allegedly taught by Helgeson) does not provide sufficient motivation that would enable one skilled in the art to make the proposed combination or modification.

Second, if the proposed modification were made (without sufficient motivation to do so), the proposed modification would render the invention of Carneal unsatisfactory for its intended purpose. If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

An intended purpose of Carneal is to reduce a delay associated with the retrieval of inline objects of web pages retrieved over a satellite link (e.g., a wireless link). *See*, Carneal, Abstract. To accomplish this purpose, Carneal utilizes a client-side proxy server (68, Fig. 6) including a satellite gateway (72) and an access point (70) connected via a wireless link (62). Prior to, during or after transmission of the parent file from the satellite gateway to the access point, the satellite gateway functions to parse the parent file, extract the external reference to the inline object, and forward a request over the Internet to prefetch the inline object. Once received, the access point stores the inline object for later reference in object cache 71 (*See*, Carneal, col. 8, lines 29-42). By placing the object cache within the access point (in client-side proxy server), inline objects can be prefetched from the Internet and, when needed, can be promptly received by web browser 20, thus, "reducing the effect of the round-trip delay associated with transversal of the satellite link." (Carneal, col. 7, line 66 to col. 8, line 1).

If the teachings of Carneal were modified so as to store the prefetched inline objects in a cache located within a web server (as suggested by the Examiner), instead of the client-side proxy server, the extra delay associated with the retrieval of inline objects over a satellite link (such as wireless link 62) would not be reduced. Since the modifications proposed by the Examiner would render the invention of Carneal unsatisfactory for its intended purpose, such modifications cannot be made.

Helgeson is merely relied upon to show that caching can be performed on the web server side. Since Helgeson provides no other teachings relevant to the presently claimed case, the combination of Helgeson and Carneal cannot be relied upon to render the above-mentioned limitations of present claim 1 unpatentable.

3. The Examiner has failed to adequately support and/or establish a p*rima facie* ground of obviousness.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all claim limitations. MPEP § 2143. None of these three criteria have been met by the Examiner in the present case. First of all, no suggestion or motivation to modify the cited references can be found within the cited references to teach or suggest the aforementioned limitation of claim 1, as explained above in Argument 2. The criterion of a reasonable expectation of success cannot be met if no teaching, suggestion or motivation exists, because there is then nothing at which to be successful. Finally, none of the cited art, either alone or in combination, teaches all of the limitations of claim 1, as explained above in Arguments 1 and 2. The third criterion recited above has therefore also not been met, and a *prima facie* case of obviousness has not been established.

Conclusion

As explained in Arguments 1-3 above, at least some limitations of claim 1, and therefore at least some limitations of claims 2, 3 and 5-8, are not taught or suggested by the cited art. Furthermore, there is no teaching, suggestion or motivation to modify the cited art to teach the limitations of these claims. For at least the reasons set forth above, claims 1-3 and 5-8 are patentably distinct over the cited art, and the rejection of Group I claims 1-3 and 5-8 under 35 U.S.C. § 103(a) is therefore asserted to be erroneous.

Patentability of Group II Claims 9-15, 17 and 18:

1. **Carneal and Helgeson each fail to provide teaching, suggestion or motivation for a method for caching a parent and child server page, where the method includes storing code for the parent server page in a cache located internal to a web server.**

Independent claim 9 recites, in part, "[a] method for caching a parent and a child server page, comprising: storing code for the parent server page in a cache located internal to a web server." Independent claim 17 (a computer program product) and independent claim 18 (a web server) recite similar limitations. As set forth above in the arguments for the patentability of Group I claims, Carneal and Helgeson fail, both separately and in combination, to provide teaching, suggestion or motivation for a cache located within a web server and containing code for a parent server page and a child server page. For at least this reason, Carneal and Helgeson cannot be relied upon to provide teaching, suggestion or motivation for a method (claim 9), computer program product (claim 17) or web server (claim 18) that stores code for a parent server page in a cache located internal to a web server.

2. **Carneal and Helgeson each fail to provide teaching, suggestion or motivation for a method for caching a parent and child server page, where the method includes storing only one copy of the code for the child server page in the cache.**

Independent claim 9 recites, in part, "[a] method for caching a parent and a child server page, comprising... storing only one copy of the code for the child server page in the cache." Independent claim 17 (a computer program product) and independent claim 18 (a web server) recite similar limitations. In this manner, the Specification and present claims provide a means for improving cache capacity by requiring that only one copy of the child server page be cached, regardless of the number of times it is called in the parent server page.

Statements in the Final Office Action suggest that Carneal discloses a method for caching parent and child server pages, where the method includes "storing only one copy (e.g., col. 6, lines 36-44, an inline object is a child page) of the code for the child server page in the cache" (Final Office Action, page 5). Though Carneal states that prefetching "involves caching documents that are likely to be requested by a client" in the above-referenced passage (col. 6, lines 36-44), Carneal is silent as to the number of copies of child server page code that may be stored within the cache. One simply cannot assume that Carneal intends to store only one copy of a child server page, as presently claimed, when Carneal lacks explicit or implicit teaching, suggestion or even motivation for doing so.

As noted above, Helgeson is merely relied upon to show that caching can be performed on the web server side. Since Helgeson provides no other teachings relevant to the presently claimed case, the combination of Helgeson and Carneal cannot be relied upon to render the above-mentioned limitation of present claims 9, 17, and 18 unpatentable.

3. The Examiner has failed to adequately support and/or establish a p*rima facie* ground of obviousness.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all claim limitations. MPEP § 2143. None of these three criteria have been met by the Examiner in the present case. First of all, no suggestion or motivation to modify the cited references can be found within the cited references to teach or suggest the aforementioned limitation of claims 9, 17, and 18, as explained above in Arguments 1 and 2. The criterion of a reasonable expectation of success cannot be met if no teaching, suggestion or motivation exists, because there is then nothing at which to be successful. Finally, none of the cited art, either alone or in combination, teaches all of the limitations of claims 9, 17, and 18, as explained above in Arguments 1 and 2. The third criterion recited above has therefore also not been met, and a *prima facie* case of obviousness has not been established.

Conclusion

As explained in Arguments 1-3 above, at least some limitations of claims 9, 17 and 18, and therefore at least some limitations of claims 10-15, are not taught or suggested by the cited art. Furthermore, there is no teaching, suggestion or motivation to modify the cited art to teach the limitations of these claims. For at least the reasons set forth above, claims 9-16, 17, and 18 are patentably distinct over the cited art, and the rejection of Group II claims 9-16, 17, and 18 under 35 U.S.C. § 103(a) is therefore asserted to be erroneous.

IX. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-3, 5-15, 17, and 18 was erroneous, and reversal of the Examiner's decision is respectfully requested.

The Commissioner is hereby authorized to charge the required fee(s) to Daffer McDaniel, LLP deposit account 50-3268/5468-05300.

Respectfully submitted,

Kevin L. Daffer
Reg. No. 34,146
Attorney for Appellant

Daffer McDaniel, LLP
P.O. Box 684908
Austin, TX 78768-4908
Date: January 18, 2005
JMF

X. APPENDIX

The present claims on appeal are as follows.

1. A software system supporting distributed web applications, comprising:

a parent server page, containing a call to a child server page;

a cache within a web server, containing code for the parent server page and child server page, wherein the code for the parent server page does not contain all the code for the child server page; and

a link associated with the call to the child server page, and encapsulating information for locating and executing the code for the child server page.

2. The software system as recited in claim 1, wherein the child server page may be executed using the link, without executing the parent server page.

3. The software system as recited in claim 1, wherein the link further comprises a web page address and a list of request attributes.

5. The software system as recited in claim 1, further comprising an instruction sequence that may be invoked to locate the child server page in the cache.

6. The software system as recited in claim 1, further comprising an object-oriented software system.

7. The software system as recited in claim 1, wherein a server page comprises a Java Server Page (JSP).

8. The software system as recited in claim 7, wherein the child JSP may be executed in response to a request made to the web server by a client or another web server.

9. A method for caching a parent and a child server page, comprising:

storing code for the parent server page in a cache located internal to a web server, such that the code for the parent server page does not contain all lines of code for the child server page;

storing only one copy of the code for the child server page in the cache;

creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.

10. The method as recited in claim 9, wherein a server page comprises a Java Server Page (JSP).

11. The method as recited in claim 10, further comprising invoking an instruction sequence to locate the code for the child JSP in the cache, in response to a request made by a web browser.

12. The method as recited in claim 10, further comprising executing the code for the child JSP using the link, without executing all the code for the parent JSP.

13. The method as recited in claim 12, wherein the child JSP is executed in the web server in response to a request made by a client or another web server.

14. The method as recited in claim 13, wherein the child JSP is executed only if it cannot first be located in the cache.

15. The method as recited in claim 13, wherein the cached child JSP may be updated without also updating the parent JSP.

17. A computer program product in a computer readable medium for use in storing a parent and a child server page in a cache, the computer program product comprising:

instructions for storing code for the parent server page in the cache located within a web server, such that the code for the parent server page does not contain all lines of code for the child server page;

instructions for storing only one copy of the code for the child server page in the cache;

instructions for creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

instructions for associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.

18. A web server including memory and processor comprising;

means for storing code for the parent server page in a cache located within the web server such that the code for the parent server page does not contain all lines of code for the child server page;

means for storing only one copy of the code for the child server page in the cache;

means for creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

means for associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.